DEFINING FLEXIBLE PROBLEM_THRESHOLDS WITH THE NEW TRIGGER SYNTAX

SERGEY SIMONENKO
Technical Support Engineer, Zabbix, Latvia
NEW SYNTAX AND FUNCTIONS

ONE SYNTAX FOR EVERYTHING
Calculated items and triggers share one syntax:

\[ \text{avg}(/\text{host/system.cpu.util},1h) \] – this is a calculated item

\[ \text{avg}(/\text{host/system.cpu.util},1h)>25 \] – a trigger

Most functions can be used both in calculated items and triggers

No worries: the conversion from the old syntax to the new one is performed automatically during upgrade
SMART PARAMETERS

- No longer necessary to pass host and item to every function. Only history and prediction functions require them *(always as first parameter)*:

  last(/host/item)=“success”

  But: just time()>=090000 instead of old {HOST:ITEM}.now()>=090000

- **Note**: still necessary to use at least one host/item reference in expression
TIME AND TIME SHIFT

- Time and time shift parameters are now one parameter:
  \((\text{sec} | \#\text{num})<:\text{time shift}>\)

- Examples:
  - \(1\text{d}:\text{now}/d\) Yesterday
  - \(1\text{d}:\text{now}/d+1\text{d}\) Today
  - \(2\text{d}:\text{now}/d+1\text{d}\) Last 2 days
  - \(1\text{w}:\text{now}/w\) Last week
  - \(1\text{w}:\text{now}/w+1\text{w}\) This week
NESTED FUNCTIONS

Nested functions are now possible, for example:

\[
\begin{align*}
\text{abs} & (\text{last}(/\text{host/item})) \\
\text{length} & (\text{find}(/\text{host/item}, \text{"pattern"})) \\
\text{round} & (\text{max}(/\text{host/counter}, 1\text{h}))
\end{align*}
\]

Redundant functions (abschange, strlen) were removed
NEW TRIGGER FUNCTIONS

- **History** functions – operate on historical data
- **Aggregate** functions – allow to sum, find minimum and maximum, etc.
- **Operator** functions – enable you to write **compact** and **better** readable expressions
- **Mathematical** functions
- **Date and time** functions (**date**, **now**, **time**, etc.)
# New Trigger Functions

## Condition

<table>
<thead>
<tr>
<th>* Item</th>
<th>Zabbix server: Available memory</th>
</tr>
</thead>
</table>

## Function

<table>
<thead>
<tr>
<th>Last of (T)</th>
<th>Date and time functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Result</td>
<td>date() - Current date</td>
</tr>
<tr>
<td></td>
<td>dayofmonth() - Day of month</td>
</tr>
<tr>
<td></td>
<td>dayofweek() - Day of week</td>
</tr>
<tr>
<td></td>
<td>now() - Number of seconds since the Epoch</td>
</tr>
<tr>
<td></td>
<td>time() - Current time</td>
</tr>
</tbody>
</table>

## History functions

| change() - Difference between last and previous value |
| changecount() - Number of changes between adjacent values, Mode (all - all changes, inc - only increases, dec - only decreases) |
| first() - The oldest value in the specified time interval |
| fuzzytime() - Difference between item value (as timestamp) and Zabbix server timestamp is less than or equal to T seconds (1 - true, 0 - false) |
| * Result |
| last() - Last (most recent) T value |
| monodec() - Check for continuous item value decrease (1 - data is monotonic, 0 - otherwise), Mode (strict - require strict monotonicity) |
NEW STRING AND MATH FUNCTIONS

- left, right, mid – character(s) at given position (index)
  - insert, replace, concat
  - trim, ltrim, rtrim
  - ascii, bitlength, bytelen

- cos, sin, sqrt, log, mod, pi, rand and many more...
OPERATOR FUNCTIONS

- Enable you to write expressions which are **compact** and **better readable**

- Before: `{HOST:ITEM.last()} >= 1 and {HOST:ITEM.last()} <= 10
  Now: **`between(last(/host/item),1,10)=1`**

- Before: `{HOST:ITEM.last()}=1 or {HOST:ITEM.last()}=2 or {HOST:ITEM.last()}=3...
  Now: **`in(last(/host/item),1,2,3,...)=1`**
NEW HISTORY AND AGGREGATE FUNCTIONS

- **monoinc, monodec** – detect monotonic increase or decrease in a set of historical values
- **changecount** – count the number of changes (all changes or only increases or decreases) between adjacent historical values
- Functions to easily work with Prometheus-compatible emitters data format: rate, bucket_percentile, histogram_quantile
ALSO

- Redundant “shortcut” functions were removed for **easier navigation** and to avoid confusion:

  - Instead of **delta** use:
    
    $$\max(/host/item, #100) - \min(/host/item, #100)$$

  - Instead of **diff** use:
    
    $$\text{last}(/host/item) \neq \text{last}(/host/item, #2)$$

  - Instead of **prev** use:
    
    $$\text{last}(/host/item, #2)$$
02

AGGREGATE CALCULATIONS

REMOVING LIMITATIONS
AGGREGATE CALCULATIONS

- Aggregate checks are now part of calculated items
- Old syntax only allowed to perform aggregate calculations based on one host group and exact item key:
  \[ \text{grpsum["MySQL-Servers","vfs.fs.size[/,total]",last]} \]
- Complex filters and wildcards introduced to address this issue
- This was a top-voted feature-request from Zabbix community
AGGREGATE CALCULATIONS

- New syntax is not limited to a single host group for aggregate calculations, you can use tags, multiple hosts groups and complex and/or logical operations with multiple clauses.

- Would like to calculate the average CPU load on a certain set of servers?

  \[
  \text{avg(last_foreach(*/system.cpu.load?[group="Servers A" or group="Servers B" or (group="Servers C" and tag="Importance:High")]))}
  \]
AGGREGATE CALCULATIONS

- New syntax supports **wildcards** when referencing items in aggregate calculations:
- Want to calculate the total traffic consumed by a customer on multiple hosts and multiple network interfaces?

\[
\text{sum(last_foreach(}/*\text{/net.if.in}[*\text{,bytes}]\text{?[group="Customer A"]})\text{)}
\]
Thank you!

www.zabbix.com